LSTM ARCHITECTURE

1. LSTM is a variant of RNN that has memory and **REMEMBERS** values over time-steps.
2. The inputs that it remembers in the previous stages are used in making a decision while generating outputs in the future stages.

Example -

“The guests who ate their stomach full even before the commencement of the event are looking for beverages. “

3. In the above example the LSTM has to remember that the word “guests” that it encountered in the previous time step is plural for it to make a decision on a later time step to generate a singular or plural verb (is/are).

4. Now, coming to the architecture which allows LSTM to **REMEMBER :-**

* It has **basic RNN cell** architecture of hidden activation layer and an output layer, and takes input from the previous stage and input from the sequence at that time-step.
* Along with the basic RNN architecture , it has **3 gates**(update gate, forget gate, output gate) to **REMEMBER** information about the past.
* The gates are **sigmoid functions** that generate values between **0-1** and all of them receive common inputs h(t-1),xt.
* While there’s only one internal state in GRU at a single time step to capture the past info and maintain it, there are **2 internal states ht and ct** at time step t of the LSTM where ht is the input from the previous time step and ct has the info about the words in the previous time steps.
* Now the **3 gates** regulate the internal state that’s responsible for memory (i.e., ct) as it moves forward through the future time-steps, based on the inputs(xt) and previous values(ht) at each time step (value of ———> sigmoid(whx.xt+whh.ht+bt) ) .
* The whole LSTM Architecture is designed in such a way that it’s more general and efficient than GRU.